

Reference [3]



⑯ BUNDESREPUBLIK
DEUTSCHLAND

⑫ Offenlegungsschrift
⑩ DE 196 35 429 A 1

⑤1) Int. Cl.⁶:
G 06 F 17/30



DEUTSCHES
PATENTAMT

≡ US 6,510,435

DE 196 35 429 A 1

⑦ Anmelder:
Bayer, Rudolf, Prof., 82194 Gröbenzell, DE

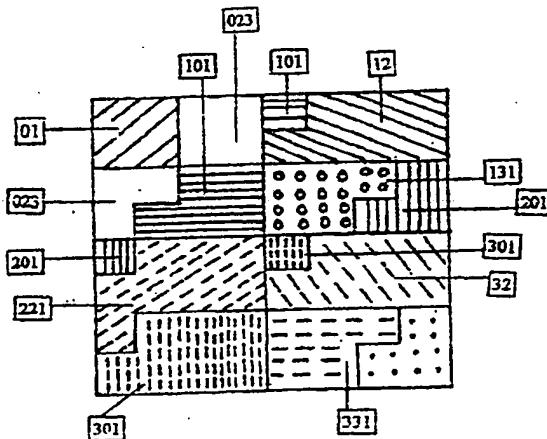
74) Vertreter:
Patentanwälte Raeck & Hössle, 70182 Stuttgart

72) Erfinder:
gleich Anmelder
55) Entgegenhaltungen:
DE 39 08 684 A1

Prüfungsantrag gem. § 44 PatG ist gestellt

② Datenbanksystem und Verfahren zum Verwalten eines n-dimensionalen Datenbestands

54) Datenbanksystem und Verfahren Lern. 10
55) Datenbanksystem und Verfahren zum Verwalten eines multidimensionalen Datenbestands. Das Datenbanksystem umfasst eine Recheneinrichtung, einen Hauptspeicher und eine insbesondere periphere Speicherseinrichtung, wobei zum Indizieren und Speichern des in einem mehrdimensionalen Würfel liegenden Datenbestands auf Speicherseiten gegebener Speicherkapazität das peripheren Speichermittels ein wiederholtes iteratives Unterteilen des mehrdimensionalen Würfels in allen Dimensionen in Subwürfel erfolgt, bis aufeinanderfolgende Subwürfel zu Regionen zusammenfaßbar sind, die jeweils eine Menge von Datenobjekten beinhalten, die auf einer der Speicherseiten gegebener Speicherkapazität abspeicherbar sind. Mit den erfundsgemäßem Verfahren zum Verwalten, Einfügen, Löschen und Suchen von Datenobjekten wird eine als FB-Baum bezeichnete dynamische Datenstruktur bereitgestellt, mit deren Hilfe verbesserte Zugriffzeiten erzielt werden und die somit zur Verwendung in Online-Anwendungen geeignet ist.



DE 19635429 A1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen
SUNDESDRUCKEREI 01. 98 702 070/437

Beschreibung

Die vorliegende Erfindung betrifft ein Datenbanksystem sowie ein Verfahren zum Verwalten eines in einem n -dimensionalen Würfel mit $n > 1$ liegenden Datenbestands.

5 Zur Verwaltung umfangreicher eindimensionalen Datenmengen auf Massenspeichern, wie beispielsweise Magnetplattenspeichern ist als Datenstruktur der sogenannte B-Baum (oder auch B*-Baum oder Präfix-B-Baum) bekannt. Die Datenstruktur des B-Baums hat gegenüber derjenigen eines einfachen Suchbaums den Vorteil, daß beim Datenzugriff geringere Suchzeiten notwendig sind. Die Suchzeit, die sich zum Auffinden gewisser Daten ergibt, beträgt bei einem einfachen Suchbaum mit n Knoten mindestens $\log_2(n)$ Schritte. Bei einem Suchbaum mit 1.000.000 Knoten muß man daher mit $\log_2(1.000.000) \approx 20$ Plattenzugriffen rechnen.

10 Nimmt man eine mittlere Zugriffszahl von 0,1 Sek. an, so benötigt die Suche eines Knotens 2 Sek. Dieser Wert ist für die Praxis zu groß. Bei der Datenstruktur des B-Baums wird die Zahl der Plattenzugriffe reduziert, indem man je Zugriff nicht einen einzelnen Knoten, sondern ein ganzes, einem Knoten zugeordnetes Teilstück von der Magnetplatte in den Hauptspeicher überträgt und innerhalb dieses Teilstücks sucht. Teilt man beispielsweise den B-Baum in Bereiche zu je sieben Knoten und überträgt mit jedem Plattenzugriff einen solchen Bereich in den Hauptspeicher, so reduziert sich die Zahl der Plattenzugriffe für die Suche eines Knotens von maximal 6 auf maximal 2. Bei 1.000.000 Knoten benötigt man somit nur noch $\log_2(1.000.000) - 7$ Zugriffe. In der Praxis unterteilt man den Suchbaum meistens in Teilbereiche der Größe $2^8 - 1$ bis $2^{10} - 1$ Knoten. Bei einer Bereichsgröße von 255 Knoten benötigt man für die Suche eines Knotens in einem Baum mit 1.000.000 Knoten $\log_2(1.000.000) \approx 2.5$ Plattenzugriffe, so daß die Suche nach einem gegebenen Wert nur noch etwa 0,3 Sek. dauert. Die Suchzeit innerhalb eines Teilbereichs mit 255 Knoten, der sich im Hauptspeicher befindet, kann gegenüber dem Plattenzugriff vernachlässigt werden. Der B-Baum ist ein höhen-balancierter Baum, bei dem alle Blätter auf dem gleichen Niveau liegen.

15 Zur Verwaltung eines multi-dimensionalen Datenbestands sind aus K. Mehlihorn: Multidimensional Searching and Computational Geometry, Springer, Heidelberg 1984, die sogenannten dd-Bäume bekannt. Mit den dd-Bäumen lassen sich prinzipiell drei Arten von Anfragen durchführen, nämlich Punkt-Anfragen, Bereichs-Anfragen und Anfragen, bei denen manche Intervalle als (-unendlich, +unendlich) angegeben sind. Die Datenstruktur eines dd-Baums gestattet jedoch nur bei Punktanfragen einen schnellen Zugriff da dann nur ein Pfad im Baum abgesucht werden muß. Bei den anderen Anfragen kann es geschehen, daß der gesamte Baum durchsucht werden muß. Darüber hinaus sind dd-Bäume statisch, d. h. die gesamte zu verwaltende Objektmenge muß schon bekannt sein, bevor der dd-Baum aufgebaut werden kann. Bei den meisten Anwendungen in der Praxis ist aber die Objektmenge dynamisch, d. h. Objekte müssen in beliebiger Reihenfolge und jederzeit im Baum eingefügt und gelöscht werden können, ohne daß der ganze Baum neu aufgebaut werden muß. Des weiteren eignen sich dd-Bäume nur für Hauptspeicheranwendungen, jedoch nicht für Peripheriespeicher, die zum Abspeichern sehr großer Datenmengen benötigt werden.

20 35 In "The Grid File" von Nievergelt et al, ACM TODS, Bd. 9, Heft 1, März 1984 sind zur Verwaltung multidimensionaler Daten sogenannte Grid-Files beschrieben, bei denen Anfragen für Punkte und Bereiche auf der Basis einer Inhaltsverzeichnis-Struktur, dem sogenannten Grid, durchgeführt werden. Obwohl diese Datenverwaltung eine schnelle Suche bei Punkt- und Bereichsanfragen gewährleistet, handelt es sich um ein statisches Verfahren, so daß beim dynamischen Einfügen und Löschen von Datenobjekten regelmäßig die gesamte Inhaltsverzeichnisstruktur völlig umorganisiert werden muß. Somit ist dieses Verfahren für viele Anwendungen, insbesondere für Online-Anwendungen nicht geeignet.

40 Aus A. Guttman: A dynamic Index Structure for spatial Searching, Proceedings ACM SIGMOD Int'l Conference on Management of Data, 1984, Seiten 47 - 57, sind als Datenstruktur zur Verwaltung multidimensionaler Daten sogenannte R-Bäume bekannt. Diese Bäume, die hauptsächlich bei sogenannten Geo-Datenbanken Anwendung finden, sind ähnlich wie B-Bäume in der Höhe balanciert und gestatten auch dynamisches Einfügen und Löschen von Objekten. Bei der Beantwortung von Anfragen sind jedoch keine schnellen Zugriffszeiten garantiert, weil unter Umständen beliebig viele Pfade in dem betreffenden Baum, im Extremfall sogar der gesamte Baum, durchsucht werden müssen, um eine Anfrage zu beantworten. Somit sind diese R-Bäume für die meisten Online-Anwendungen nicht geeignet.

45 55 Das heute in der Praxis am weitesten verbreitete Verfahren zur Verwaltung eines multidimensionalen Datenbestands basiert auf den ursprünglichen eindimensionalen B-Bäumen, wobei für jede Dimension des Ausgangsdatenbestands jeweils ein B-Baum eingesetzt wird so daß Bereichsanfragen in einem n -dimensionalen Datenbestand durch n B-Bäume unterstützt werden. Bei einer Bereichsanfrage werden somit für jede Dimension sämtliche Objekte vom peripheren Speicher geholt, deren Werte innerhalb des in der Anfrage spezifizierten Intervalls für diese Dimension liegen. Diese Datenobjekte bilden die Treffermenge in der betreffenden Dimension. Um die gewünschte Antwortmenge zu bestimmen, muß eine Schnittmenge der Treffermengen sämtlicher Dimensionen berechnet werden, was üblicherweise erst eine Sortierung dieser Mengen erfordert. Auch beim Einfügen und Löschen eines Datenobjekts müssen entsprechend n B-Bäume durchsucht und modifiziert werden.

60 65 Ausgehend hiervon liegt der Erfindung die Aufgabe zugrunde, ein Datenbanksystem und ein Verfahren zum Verwalten eines n -dimensionalen Datenbestands bereitzustellen, das aufgrund besserer Zugriffszeiten insbesondere zur Verwendung in Online-Anwendungen geeignet ist und das ein dynamisches Einfügen und Löschen von Datenobjekten gestattet.

70 Zur Lösung der Aufgabe wird erfundungsgemäß ein Datenbanksystem mit den Merkmalen des Anspruches 1 vorgeschlagen. Das erfundungsgemäß Datenbanksystem umfaßt eine Recheneinrichtung, einen Hauptspeicher und eine Speichereinrichtung, bei der es sich insbesondere um eine peripherie Speichereinrichtung handelt. Der Grundgedanke der Erfindung liegt darin, einen zu verwaltenden multidimensionalen Datenbestand in einen mehrdimensionalen Würfel zu legen und zum Indizieren und Speichern dieses Datenbestands mittels der

Recheneinrichtung ein wiederholtes iteratives Unterteilen des mehrdimensionalen Würfels in allen Dimensionen in Subwürfel vorzunehmen. Das Unterteilen wird dabei so oft wiederholt, bis aufeinanderfolgende Subwürfel zu Regionen zusammenfaßbar sind die jeweils eine Menge von Datenobjekten beinhalten, die auf einer der Speicherseiten gegebener Speicherkapazität der insbesondere peripheren Speichereinrichtung abspeicherbar sind. Da die Regionen aus aufeinanderfolgenden Subwürfeln zusammengefaßt sind, liegen auch die Regionen hintereinander, so daß sie eine eindimensionale Struktur bilden. Somit ist erfahrungsgemäß bei einem Einfügen bzw. Löschen von Datenobjekten lediglich die Modifikation einer einzigen Datenstruktur, beispielsweise eines Baumes, erforderlich.

In Ausgestaltung der Erfindung erfolgt das Abspeichern der Datenobjekte einer Region auf einer Speicherseite gegebener Speicherkapazität unter Zuordnung eines Zeigers zu der Speicherseite und einer die Regionengrenzen bezeichnenden Adresse. Somit sind jeder abzuspeichernde Region eindeutige, die Regionengrenzen definierende Adressen sowie ein auf die Speicherseite, auf welcher die betreffende Region abgespeichert ist, verweisender Zeiger zugeordnet. Dadurch wird das Auffinden der Region und der in der Region enthaltenen Datenobjekte bei Verwaltungsvorgängen wie Beantwortung von Anfragen, Löschen oder Einfügen von Datenobjekten erleichtert.

In weiterer Ausgestaltung der Erfindung erfolgt die Speicherung des Zeigers und der Adresse in einem B-Baum, B*-Baum oder Präfix-B-Baum, so daß bei einer Adresssuche eine einfache und schnell durchzuführende Suche in einem B-Baum zur Identifikation der benötigten Region über den der Adresse zugeordneten und auf die Speicherseite der benötigten Region verweisenden Zeiger erfolgen kann.

In weiterer Ausgestaltung der Erfindung erfolgt die Speicherung der Datenobjekte selbst in den Blattseiten des B-Baums, B*-Baums oder Präfix-B-Baums.

In vorteilhafter Ausgestaltung der Erfindung besteht die die Regionengrenzen bezeichnende Adresse aus Angaben über den letzten der die Region bildenden Subwürfeln. Besonders vorteilhaft erweist sich ein Datenbanksystem, bei dem die Adresse Angaben über die Anzahl der auf jeder Unterteilungsstufe in der Region enthaltenen Subwürfel umfaßt. Eine Region ist somit eindeutig bezeichnet, wenn der letzte vollständig in der Region enthaltene Subwürfel durch die Adress-Angabe ebenfalls eindeutig bezeichnet ist. Der Beginn der Region wird dabei durch die Adress-Angabe über den letzten der die vorhergehende Region bildenden Subwürfeln angegeben.

Zur weiteren Lösung der der Erfindung zugrundeliegenden Aufgabe wird ein Verfahren zum Verwalten eines in einem n -dimensionalen Würfel mit $n > 1$ liegenden Datenbestands mit den Merkmalen des Anspruchs 8 vorgeschlagen. Mit dem erfahrungsgemäßen Verfahren wird zum Indizieren und Speichern eines multidimensionalen Datenbestands dieser Datenbestand in einen n -dimensionalen Würfel mit $n > 1$ gelegt. Dieser Würfel bildet in seiner Gesamtheit eine Ausgangsregion, die sämtliche Datenobjekte des Datenbestands enthält. Sollte die Anzahl der vorhandenen Datenobjekte kleiner oder gleich sein als die der vorgegebenen Speicherkapazität einer Speicherseite entsprechende Zahl von Datenobjekten, so wird die Ausgangsregion auf einer Speicherseite abgespeichert. Andernfalls wird die Ausgangsregion entlang einer Spaltadresse gespalten, wobei die Spaltadresse so gewählt wird, daß etwa entlang der Datenmitte zwei neue Teilregionen erzeugt werden. Mit jeder dieser Teilregionen wird anschließend so verfahren, wie zuvor mit der Ausgangsregion, d. h. es wird die Anzahl der jeweils in der Teilregion enthaltenen Datenobjekte bestimmt und mit der der vorgegebenen Speicherkapazität einer Speicherseite entsprechenden Zahl verglichen. Ist der Datenbestand nicht größer als die der vorgegebenen Speicherkapazität entsprechende Zahl, so wird die betreffende Region auf einer Speicherseite abgespeichert, andernfalls wird sie wiederum entlang der Datenmitte gespalten und das Verfahren beginnt von neuem.

Vorteilhafterweise erfolgt das Abspeichern der Datenobjekte einer Region oder Teilregion unter paralleler Abspeicherung einer der betreffenden Region zugeordneten Adresse sowie eines der Adresse zugeordneten und auf die Speicherseite, auf der die abgespeicherten Datenobjekte enthalten sind, verweisenden Zeigers. Bei der parallel abzuspeichernden Adresse kann es sich vorteilhafterweise um die Spaltadresse handeln, die das Ende der einen und den Anfang der anderen Region angibt.

In Ausgestaltung der Erfindung erfolgt die Speicherung der Adresse und des Zeigers in einem B-Baum, B*-Baum oder Präfix-B-Baum, wobei durch aufeinanderfolgende Adressen jeweils Regionen definiert werden, deren Datenobjekte jeweils auf einer Speicherseite gegebener Speicherkapazität abgespeichert sind.

Zur weiteren Lösung der der Erfindung zugrundeliegenden Aufgabe wird des weiteren ein Verfahren zum Einfügen von Datenobjekten in einen in Regionen aufgeteilten und auf Speicherseiten gegebener Speicherkapazität einer insbesondere peripheren Speichereinrichtung abgespeicherten n -dimensionalen Datenbestand mit $n > 1$ mit den Merkmalen des Anspruchs 11 vorgeschlagen. Vorteilhafterweise handelt es sich bei dem abgespeicherten n -dimensionalen Datenbestand um einen gemäß dem vorstehend beschriebenen erfahrungsgemäßem Verfahren indizierten und abgespeicherten Datenbestand. Erfahrungsgemäß wird auschließlich von den Koordinaten des einzufügenden Datenobjekts eine das Datenobjekt enthaltende Region des n -dimensionalen Datenbestands sowie die Speicherseite, auf der diese Region abgespeichert ist, ermittelt. Daraufhin werden die auf dieser Speicherseite gespeicherten Datenobjekte gezählt. Sollte die Anzahl der gespeicherten Datenobjekte kleiner sein als die der gegebenen Speicherkapazität der Speicherseite entsprechende Anzahl, so wird das einzufügende Datenobjekt ebenfalls auf dieser Speicherseite abgespeichert. Andernfalls wird eine Spaltadresse für die auf dieser Speicherseite gespeicherte Region derart ausgewählt, daß durch Spalten der Region entlang dieser Spaltadresse eine erste und eine zweite Teilregion erzeugt werden, in welchen jeweils weniger als etwa die Hälfte der der gegebenen Speicherkapazität entsprechenden Anzahl an Datenobjekten enthalten ist. Dann wird das einzufügende Datenobjekt in diejenige Teilregion eingelegt, in welcher die Koordinaten des Datenobjekts liegen, woraufhin die erste und die zweite Teilregion auf jeweils einer Speicherseite abgespeichert werden.

Erfahrungsgemäß ist somit in der gegebenen Datenstruktur ein dynamisches Einfügen von Datenobjekten möglich, ohne daß die Gesamtheit der Datenstruktur modifiziert oder neu angelegt werden muß. Sollte durch

das Einfügen des neuen Datenobjekts die Region, in welche die Einfügung vorgenommen wurde, nicht mehr auf einer Speicherseite abspeicherbar sein, so wird diese Region in zwei weitere Regionen aufgespalten, wodurch lediglich die betreffende zu spaltende Region bzw. die durch die Spaltung neu entstandenen Teilregionen modifiziert und neu abgespeichert werden müssen.

5 Vorteilhafterweise erfolgt das Auffinden der Speicherseite in dem erfundungsgemäßen Verfahren zum Einfügen von Datenobjekten mittels in einem B-Baum, B*-Baum oder Präfix-B-Baum gespeicherter, den Speicherseiten zugeordneter Adressen und Zeiger. Dadurch gestaltet sich das Auffinden der gesuchten Speicherseite besonders einfach und schnell. Entsprechend erweist es sich als vorteilhaft, wenn das Speichern der neu entstandenen Teilregionen unter Ersetzung des vormaligen Zeigers und der Adresse der gespaltenen Region durch jeweils der ersten und der zweiten Teilregion zugeordnete Adressen und Zeiger erfolgt. Dabei kann beispielsweise für die erste Teilregion als begrenzende Adresse die Spaltadresse verwendet werden und für die zweite Teilregion kann die begrenzende Adresse der gespaltenen Region verwendet werden.

10 Als besonders vorteilhaft erweist es sich, wenn die Speicherung der Adresse und des Zeigers in einem B-Baum, B*-Baum oder Präfix-B-Baum erfolgt, wobei durch aufeinanderfolgende Adressen jeweils Regionen definiert werden, deren Datenobjekte jeweils auf einer Speicherseite gegebener Speicherkapazität abgespeichert sind.

15 Zur weiteren Lösung der der Erfundung zugrundeliegenden Aufgabe wird ein Verfahren zum Löschen von Datenobjekten mit den Merkmalen des Anspruches 15 vorgeschlagen. Danach werden ausgehend von den Koordinaten des zu löschenen Datenobjektes die das Datenobjekt enthaltende Region des n-dimensionalen Datenbestands sowie die Speicherseite, auf welcher diese Region abgespeichert ist, bestimmt und das zu löschen Objekt wird auf dieser Speicherseite gelöscht. Danach wird die Anzahl der auf dieser Speicherseite gespeicherten Datenobjekte bestimmt und die Region wird mit einer ihrer beiden Nachbarregionen verschmolzen, falls die Anzahl der gespeicherten Datenobjekte kleiner ist als etwa die Hälfte der der gegebenen Speicherkapazität der Speicherseite entsprechenden Anzahl. Daraufhin wird wiederum die Anzahl der in der durch das Verschmelzen neu entstandenen Region vorhandenen Datenobjekte bestimmt. Ist diese Anzahl kleiner als die der gegebenen Speicherkapazität einer Speicherseite entsprechende Anzahl, so wird die Region auf einer Speicherseite abgespeichert, andernfalls wird eine Spaltadresse für die Region derart ausgewählt, daß durch Spalten entlang der Spaltadresse eine erste Teilregion und eine zweite Teilregion erzeugt werden, die jeweils etwa die Hälfte der in der zu spaltenden Region enthaltenen Datenobjekte enthalten, woraufhin die entstandenen Teilregionen auf jeweils einer Speicherseite abgespeichert werden.

20 30 Vorteilhafterweise erfolgt auch bei diesem Verfahren das Auffinden der Speicherseite mittels in einem B-Baum, B*-Baum oder Präfix-B-Baum gespeicherter, den Speicherseiten zugeordneter Adressen und Zeiger.

35 Zur weiteren Lösung der der Erfundung zugrundeliegenden Aufgabe wird ein Verfahren zum Durchführen einer Datenanfrage auf der Grundlage eines gegebenen n-dimensionalen Anfragebereichs mit den Merkmalen des Anspruches 17 vorgeschlagen. Danach werden die Koordinaten des niedrigsten und des höchsten Schnittpunktes des gegebenen Anfragebereichs mit dem n-dimensionalen Datenbestand sowie diejenige Region bestimmt, in welcher der niedrigste Schnittpunkt liegt. Danach wird die Speicherseite aufgefunden, auf der die ermittelte Region abgespeichert ist, und es werden sämtliche auf dieser Speicherseite abgespeicherte Datenobjekte ermittelt, die eine Schnittmenge mit dem Anfragebereich bilden. Die ermittelten Datenobjekte werden daraufhin ausgegeben. Dann wird der in der Abfolge letzte Subwürfel der ermittelten Region bestimmt, der den Anfragebereich schneidet und die Datenanfrage wird beendet, falls der höchste Schnittpunkt des Anfragebereichs in diesem Subwürfel liegt. Andernfalls wird der nächste Subwürfel derselben Ebene und desselben übergeordneten Würfels ermittelt, der den Anfragebereich schneidet, und es werden die Koordinaten des niedrigsten Schnittpunktes des Anfragebereichs mit dem neu ermittelten Subwürfel bestimmt, woraufhin das Verfahren beim Bestimmen derjenigen Region, in welcher der niedrigste Schnittpunkt liegt, weitergeführt wird, falls ein Subwürfel ermittelt wurde. Andernfalls wird der nächste Subwürfel der Ebene des übergeordneten Würfels ermittelt, der den Anfragebereich schneidet und die Ermittlung des nächsten Subwürfels derselben Ebene und desselben übergeordneten Würfels, der den Anfragebereich schneidet, wird mit den Subwürfeln des neu ermittelten Würfels durchgeführt. Wird kein Subwürfel der Ebene des übergeordneten Würfels ermittelt, übernimmt der übergeordnete Würfel die Rolle des Subwürfels und es wird dann der nächste Subwürfel dieser Ebene und desselben übergeordneten Würfels ermittelt, der den Anfragebereich schneidet. So werden erfundungsgemäß sukzessive die Subwürfel aller relevanten übergeordneten Würfel und wiederum derer übergeordneter Würfel bezüglich Schnittmengen von Datenobjekten mit dem Anfragebereich überprüft.

40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9

Fig. 10 zeigt den Abfragebereich der Fig. 3 und 9 mit mehreren in dem Abfragebereich liegenden und den Abfragebereich schneidenden Subwürfeln.

Fig. 11 zeigt einen Datenbestand am Beispiel eines ausgedehnten Objekts.

Zur Verwaltung eines n -dimensionalen Datenbestands wird erfundungsgemäß davon ausgegangen, daß der Datenraum, in dem sich die zu verwaltenden Datenobjekte befinden, ein n -dimensionaler Würfel ist oder von einem solchen Würfel eingeschlossen wird, wobei n eine beliebige natürliche Zahl mit $n > 1$ ist. Dieser Würfel wird umgebender Würfel genannt.

Der umgebende Würfel wird unterteilt in 2^n Subwürfel gleicher Größe, indem jede Dimension des Würfels halbiert wird. Diese Subwürfel werden in einer festzulegenden Reihenfolge von 1 bis 2^n durchnumeriert. Fig. 1 zeigt einen 2-dimensionalen Würfel, also ein Quadrat, für den Fall eines 2-dimensionalen Datenraums, unterteilt in $2^2 = 4$ gleichgroße Subwürfel, die links oben beginnend von links nach rechts und von oben nach unten durchnumeriert sind.

Fig. 2 zeigt einen umgebenden Würfel eines 3-dimensionalen Datenraums mit $2^3 = 8$ gleichgroßen Subwürfeln die ebenfalls durchnumeriert sind, und zwar hinten links beginnend von links nach rechts, von oben nach unten und von hinten nach vorn. Demgemäß kommt der Subwürfel mit der Nummer 3 in der Zeichnung der Fig. 2 links hinten unten zu liegen und ist nicht sichtbar.

Jeder der in den Fig. 1 und 2 gezeigten Subwürfel des Ausgangswürfels kann nach demselben Verfahren wiederum in 2^n Subwürfel mit der Numerierung von 1 bis 2^n unterteilt werden und diese Unterteilung kann rekursiv (bzw. iterativ) beliebig oft fortgesetzt werden. In der praktischen Anwendung der Erfindung wird die Unterteilung solange durchgeführt, bis durch zusammenliegende Subwürfel Regionen gebildet werden können, deren Anzahl von Datenobjekten auf einer Speicherseite gegebener Speicherkapazität speicherbar sind.

Hat der Ausgangswürfel eine Seitenlänge l , so haben die Subwürfel nach s Unterteilungen noch die Seitenlänge $(1/2^s)l$. Zur Identifizierung der Würfel werden diese entsprechend der jeweiligen Unterteilung in eine Stufe s eingeordnet. Demgemäß hat der Ausgangswürfel die Stufe 0 die in den Fig. 1 und 2 dargestellten Subwürfel haben die Stufe 1 usw.

Eine Area A ist nun ein spezieller Unterraum des Ausgangswürfels, der wie folgt entsteht:

- auf der Stufe 1 gehören die ersten a_1 Subwürfel vollständig zu der Area A, wobei $0 < a_1 < 2^n$ ist
- auf der Stufe 2 gehören die ersten a_2 Subwürfel des Subwürfels $a_1 + 1$ der ersten Stufe zu der Area A
- usw. bis zur Stufe i , wo die ersten a_i Subwürfel des Subwürfels $a_{i-1} + 1$ der i -ten Stufe zu der Area A gehören.

Eine auf diese Weise definierte Area A ist eindeutig beschrieben durch die Folge von Zahlen $a_1 a_2 a_3 \dots a_i$. Diese Zahlenfolge wird die eindeutige Adresse alpha (A) der Area A genannt. Dies ist anhand der Fig. 3.1 bis 3.4 für den 2-dimensionalen Fall, also $n = 2$, veranschaulicht.

Fig. 3.1 zeigt einen 2-dimensionalen Würfel, der in vier gleichgroße Subwürfel unterteilt ist, die jeweils wiederum zweimal unterteilt sind. Die in der Fig. 3.1 grau unterlegte Fläche bildet eine Area A, welche die Adresse alpha (A) = 03 hat, weil der erste 2-dimensionale Subwürfel auf der Stufe 1 nicht vollständig zu A gehört, was durch die Ziffer 0 an der ersten Stelle der Adresse angegeben ist. Von diesem Subwürfel gehören dann aber die ersten 3 Subwürfel der Stufe 2 zu A, was durch die Ziffer 3 an der zweiten Stelle der Adresse angegeben wird.

Fig. 3.2 zeigt den Würfel der Fig. 3.1 mit einer anderen grau unterlegten Fläche, die die Area B bildet. Diese hat die Adresse alpha (B) = 132, da der erste Subwürfel komplett in der Area B enthalten ist (Ziffer 1 an der ersten Stelle), von dem zweiten Subwürfel jedoch nur die ersten drei Subwürfel der Stufe 2 enthalten sind (Ziffer 3 an der zweiten Stelle) und von dem vierten Subwürfel der Stufe 2 wiederum nur die ersten beiden Subwürfel der Stufe 3 enthalten sind (Ziffer 2 an der dritten Stelle).

Fig. 3.3 zeigt wiederum den 2-dimensionalen Subwürfel der Fig. 3.1 und 3.2 mit einer unterschiedlichen grau unterlegten Fläche, die die Area C bildet, welche die Adresse alpha (C) = 2331 hat.

In Fig. 3.4 bildet der gesamte (grau unterlegte) 2-dimensionale Würfel eine Area D mit der Adresse alpha (D) = 4. Diesem Spezialfall wird als Adresse beispielsweise epsilon zugeordnet (alpha (D) = epsilon).

Zur weiteren Veranschaulichung der Adresszuweisung ist in Fig. 4 ein eine Area E bildender Ausschnitt aus einem 3-dimensionalen Datenbestands-Würfel dargestellt, der die Adresse alpha (E) = 541 hat.

Die Subwürfel, die noch zu einer Area gehören, werden auf jeder Unterteilungsstufe exponentiell um den Faktor 2^n kleiner. Dadurch bleiben die Adressen sehr kurz. Bei einer Umsetzung beispielsweise einer Landkarte des Bundeslands Bayern hat eine Area, deren kleinster Subwürfel 8×8 Meter groß ist, eine Adresse von etwa 32 Bits Länge.

Die beschriebenen Areas sind streng linear nach ihrem mengentheoretischen Enthaltein geordnet: Für eine Area A, die raumäßig in einer Area B enthalten ist, kann man schreiben:

A enthalten in B

Des weiteren sind die zu den Areas gehörigen Adressen lexikographisch wie Wörter über einem Alphabet geordnet. Ist beispielsweise eine Adresse α kleiner als Adresse β , dann kann die bezeichnet werden durch

$\alpha < \beta$

So gilt beispielsweise

132 < 2331 und 2331 < 32

Die vorstehend beschriebenen Areas und Adressen sind nun so aufgebaut, daß folgender Zusammenhang gilt:

5 Area(α) enthalten in Area(β) genau dann wenn $\alpha < \beta$

Da die Areas wie erläutert linear geordnet sind, kann immer die Differenz zwischen der größeren und der kleineren Area gebildet werden. Ist die Adresse α kleiner als die Adresse β , also $\alpha < \beta$, dann wird eine Region $\text{reg}(\alpha, \beta)$ als die Differenz zwischen Area(β) und Area(α) definiert. Dies ist gleichbedeutend mit:

10 $\text{reg}(\alpha, \beta) = \text{Area}(\beta) - \text{Area}(\alpha)$

Regionen haben die Eigenschaft, daß sie im gegebenen n-dimensionalen Raum nach sehr speziellen Mustern geclustert sind. Fig. 7 zeigt ein Beispiel einer derartigen Clusterung eines 2-dimensionalen Würfels in mehrere Regionen. So umfaßt die erste, mit 01 bezeichnete Region exakt den ersten Subwürfel des ersten Subwürfels des Gesamtwürfels. Die zweite Region mit 023 bezeichnet, umfaßt den kompletten zweiten Subwürfel des ersten Subwürfels und die drei ersten Subwürfel des folgenden dritten Subwürfels. In der Fig. 7 ist diese Region weiß dargestellt. Die Region beginnt somit nach dem Subwürfel mit der Adresse 01 und endet mit dem Subwürfel der Adresse 023. Durch die Angabe dieser beiden Adressen ist somit gemäß der vorstehend erläuterten Vorschrift, nämlich $\text{reg}(01, 023) = \text{Area}(023) - \text{Area}(01)$, eindeutig definiert.

Die weiteren in dem Subwürfel der Fig. 7 eingezeichneten Regionen werden entsprechend unter Verwendung der in den einzelnen Regionen angegebenen Adressen gebildet. Dabei kann es durchaus vorkommen, daß Subwürfel, die eine zusammenhängende Region bilden, aufgrund der Nummerierungs- und Darstellungsweise als nicht zusammenhängend erscheinen. Dies ist genau dann der Fall, wenn eine Region sich aus Subwürfeln zusammensetzt, bei denen in der Numerierung ein Sprung von 2 nach 3 oder von 4 nach 1 vorliegt. In der Fig. 7 ist dies beispielsweise bei der schon erläuterten weiß eingezeichneten Region $\text{reg}(01, 023)$ der Fall, da dort der Subwürfel 02 und Teile des Subwürfels 03 eine Region bilden, durch den Übergang der Numerierung von 02 nach 03 jedoch in der Darstellung ein Sprung vorliegt. Dies ist u. a. auch in der sich an die Region $\text{reg}(01, 023)$ anschließenden Region $\text{reg}(023, 101)$ der Fall, die sich von dem Subwürfel 04 in den Subwürfel 11 erstreckt.

Die beschriebenen Regionen eines n-dimensionalen Würfels spielen eine zentrale Rolle bei der Speicherung von Objekten auf einem insbesondere peripheren Computerspeicher. Diese Speicher sind unterteilt in sogenannte Seiten, deren Inhalt bei einem Ein/Ausgabevorgang mit einem Speicherzugriff in den Arbeitsspeicher des Rechners geholt bzw. von dort wieder in den peripheren Computerspeicher zurückgeschrieben wird. Die Regionen werden im folgenden so konstruiert, daß die abzuspeichernden Datenobjekte, die in einer Region liegen bzw. sie schneiden, auf einer Seite des peripheren Computerspeichers gespeichert werden können. Die zur Region $\text{reg}(\alpha, \beta)$ zugehörige Speicherseite kann beispielsweise mit $\text{page}(\alpha, \beta)$ bezeichnet werden.

Bei allen Computeranwendungen spielt die Auflösung des Raumes, d. h. die kleinsten noch unterscheidbaren Raumelemente eine wesentliche Rolle. Sie werden im 2-dimensionalen Fall auch pixel (für picture element) und im n-dimensionalen Fall voxel (für volume element) genannt. Die Anzahl der Elemente, die pro Dimension unterschieden werden können, kann mit pix bezeichnet werden. Sind (x_1, x_2, \dots, x_n) die kartesischen Koordinaten eines Punkts im n-dimensionalen Raum, dann gilt $0 <= x_i < \text{pix}$ für $i = 1, 2, \dots, n$.

Die Area, deren letzter Subwürfel gerade der Punkt (x_1, x_2, \dots, x_n) ist, ist eindeutig definiert und hat eine bestimmte Adresse, die sich aus (x_1, x_2, \dots, x_n) leicht und eindeutig berechnen läßt. Wir bezeichnen diese Funktion bzw. Berechnungsvorschrift mit $\alpha(\text{pix}(x_1, x_2, \dots, x_n))$. Umgekehrt lassen sich aus der Adresse α einer Area die kartesischen Koordinaten des letzten Punktes berechnen, der noch zu dieser Area gehört. Wir bezeichnen diese Funktion mit $\text{cart}(\alpha)$. α und cart sind inverse Funktionen zueinander, d. h. es gilt:

50 $\text{cart}(\alpha(\text{pix}(x_1, x_2, \dots, x_n))) = (x_1, x_2, \dots, x_n)$
51 $\alpha(\text{cart}(\alpha)) = \alpha$

Wie beschrieben wird bei der erfundungsgemäßen Datenverwaltung ein n-dimensionaler Raum in Form eines Würfels durch eine Menge von Areas vollständig in Regionen partitioniert, wobei die Adressen der Areas lexikographisch sortiert werden und eine Region gerade durch die Differenz zweier aufeinanderfolgender Areas bzw. durch deren Adressen definiert wird. Die kleinste mögliche Area ist das kleinste Pixel des "Datenuniversums" und hat die Adresse 00 ... 01. Diese Adresse wird hier mit σ bezeichnet. Die größte Adresse ist 1 im 2-dimensionalen Fall und 2^n im n-dimensionalen Fall und wird hier, wie vorstehend schon ausgeführt mit ϵ bezeichnet.

Die sortierten Adressen der Areas werden erfundungsgemäß in einem konventionellen B-Baum, B*-Baum oder Präfix-B-Baum abgelegt. Vorteilhafterweise wird im B-Baum zwischen zwei aufeinanderfolgenden Adressen α_{i-1} und α_i , welche genau die Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ definieren, ein Zeiger (auch Pointer oder Verweis genannt) auf diejenige Seite des peripheren Speichers abgelegt, auf der die Datenobjekte der Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ gespeichert sind. Dieser Zeiger wird mit p_i bezeichnet.

Fig. 5 veranschaulicht eine derartige Abspeicherung, wobei in der in der Darstellung der Fig. 5 oberen Ebene alternierend jeweils eine Adresse und ein Zeiger abgespeichert sind. Durch die beiden jeweils einem Zeiger zugeordneten Adressen sind die Grenzen einer Region gegeben, auf deren Datenobjekte der zwischen den beiden Adressen stehende Zeiger verweist. So zeigt im Beispiel der Fig. 5 der Zeiger p_i durch den eingezeichneten Pfeil auf eine Speicherseite, in welcher die Datenobjekte (hier die Identifikatoren der Datenobjekte) der durch die jeweils links und rechts des Zeigers stehenden Adressen gebildete Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ liegen.

Der hier beschriebene Fall betrifft einen B*-Baum für Adressen, bei dem die Zeiger p_i auf sogenannte Blattseiten zeigen. Auf diesen Blattseiten stehen dann die Datenobjekte selbst oder deren Identifikatoren, wobei dann die Datenobjekte selbst noch einmal auf weitere Seiten ausgelagert sind und über ihre Identifikatoren im peripheren Speicher gefunden werden können. Die vorstehend beschriebene erfundungsgemäße Datenstruktur wird im folgenden als FB-Baum bezeichnet.

Nachfolgend wird das erfundungsgemäße Verfahren am Beispiel der Verwaltung von Punktobjekten im n -dimensionalen Raum beschrieben. Ein Punktobjekt P ist durch seine kartesischen Koordinaten (x_1, x_2, \dots, x_n) gegeben. Daraus wird die Adresse $\beta = \text{alpha}(x_1, x_2, \dots, x_n)$ berechnet. Der Punkt P liegt in der eindeutig definierten Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ mit der Eigenschaft daß

$$\alpha_{j-1} < \beta < \alpha_j.$$

Diese Region wird durch eine Baumsuche im FB-Baum bestimmt, wo der Verweis P_j auf die Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ gefunden wird. Der Punkt P , d. h. sein Identifikator zusammen mit seinen Koordinaten (x_1, x_2, \dots, x_n) wird dann auf der Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ des Peripheriespeichers abgespeichert. Alternativ dazu kann nur der Identifikator des Punktes auf $\text{page}(\alpha_{j-1}, \alpha_j)$ gespeichert werden und der Punkt selbst, d. h. seine Koordinaten und sonstige Informationen über ihn, wird nochmals auf eine andere Seite ausgelagert.

Die Seiten des peripheren Speichers haben nur eine bestimmte vorgegebene Speicherkapazität und können deshalb nur eine bestimmte Anzahl M von Objekten aufnehmen. Sobald in einer Region weitere Objekte eingefügt werden sollen, die zur Region gehörende Seite aber keine Objekte mehr aufnehmen kann, muß der Inhalt der Seite auf zwei Seiten aufgeteilt werden, und die Region muß dementsprechend in zwei Regionen gespalten werden. Nachfolgend wird zunächst die Spaltung der Region beschrieben: Sei die Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ und die zugehörige Seite $\text{page}(\alpha_{j-1}, \alpha_j)$. Wegen der Definition einer Region gilt dann: $\alpha_{j-1} < \alpha_j$. Jetzt wird eine Spaltadresse β gewählt mit der Eigenschaft, daß β zwischen den beiden Arca-Adressen α_{j-1} und α_j liegt und die Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ etwa in der Mitte spaltet, d. h. daß etwa die Hälfte der Objekte in der Region $\text{reg}(\alpha_{j-1}, \beta)$, die andere Hälfte in der Region $\text{reg}(\beta, \alpha_j)$ zu liegen kommen. Dann enthalten die beiden Regionen $\text{reg}(\alpha_{j-1}, \beta)$ und $\text{reg}(\beta, \alpha_j)$ jeweils weniger als $1/2 M + \text{eps}$ Objekte, wobei eps eine kleine vorgegebene Zahl ist und z. B. etwa $1/10$ von M sein könnte.

Anschließend werden die Objekte von der Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ auf die beiden Seiten $\text{page}(\alpha_{j-1}, \beta)$ und $\text{page}(\beta, \alpha_j)$ aufgeteilt wobei eine der beiden Seiten mit der ursprünglichen Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ identisch sein kann.

Bei der Spaltung der Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ wird dabei die aus den Fig. 6.1 und 6.2 ersichtliche Modifikation des FB-Baums durchgeführt. Fig. 6.1 zeigt die Ausgangsstruktur des FB-Baums mit einer in der Zeichnungsdarstellung oberen Ebene, die die abgespeicherten Adressen α_{j-1}, α_j und α_{j+1} sowie die zwischen diesen Adressen liegenden Zeiger p_j und p_{j+1} enthält. Dabei verweist der Zeiger p_j auf die Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ und der Zeiger p_{j+1} auf die Seite $\text{page}(\alpha_{j-1}, \alpha_j)$. Nach der Spaltung der Seite $\text{page}(\alpha_{j-1}, \alpha_j)$ liegt die aus der Fig. 6.2 ersichtliche Baumstruktur vor, wobei in der in der Zeichnungsdarstellung oberen Ebene zwischen dem Zeiger p_j und der Adresse α_j die der Spaltadresse entsprechende Adresse β und der auf die neue Speicherseite $\text{page}(\beta, \alpha_j)$ verweisende Zeiger p' eingefügt wurden. Der bisherige Zeiger p_j verweist nunmehr auf die modifizierte Seite $\text{page}(\alpha_{j-1}, \beta)$ und der Zeiger p_{j+1} verweist unverändert auf die ebenfalls unveränderte Seite $\text{page}(\beta, \alpha_{j+1})$.

Durch die Einfügung von β und p' in den übergeordneten Knoten muß möglicherweise die übergeordnete Seite ebenfalls gespalten werden, wenn dort durch diese Einfügung zuviel Adress- und Zeigerdaten vorhanden sind. Diese Spaltvorgänge verlaufen dann aber genau wie bei den aus dem Stand der Technik bekannten B-Bäumen üblich. Durch diese wiederholten Spaltvorgänge beim Einfügen von Objekten in das vorhandene Datenuniversum entstehen die FB-Bäume, die ein ganz ähnliches Wachstum zeigen wie die bekannten B-Bäume.

Zur weiteren Erläuterung des Verfahrens wird nachfolgend die Löschung von Datenobjekten aus dem vorhandenen Datenuniversum und eine Anpassung der Regionen nach erfolgter Löschung beschrieben.

Soll ein Punkt-Objekt (x_1, x_2, \dots, x_n) wieder gelöscht werden so werden zunächst wie beim Einfügevorgang die Region, in der der Punkt liegt, und die zugehörige Speicherseite bestimmt. Der Punkt wird von der Seite gelöscht und verschwindet somit auch aus der Region. Sinkt dadurch die Anzahl der Objekte in der Seite unter $1/2 M - \text{eps}$, so wird die Region mit einer der beiden Nachbarregionen verschmolzen. Wenn die so entstehende Region zu viele Objekte enthält, wird sie wieder wie schon vorher beschrieben in der Mitte gespalten.

Beispielhaft könnte die Region $\text{reg}(\alpha_{i-1}, \alpha_i)$, falls sie nach einem Löschvorgang zu wenige Objekte enthält, mit der Region $\text{reg}(\alpha_i, \alpha_{i+1})$ verschmolzen werden zur Region $\text{reg}(\alpha_{i-1}, \alpha_{i+1})$. Falls $\text{reg}(\alpha_{i-1}, \alpha_{i+1})$ dann zu viele Objekte enthält, wird sie wieder gespalten in $\text{reg}(\alpha_{i-1}, \beta)$ und $\text{reg}(\beta, \alpha_{i+1})$, wobei natürlich β geeignet gewählt wird und gelten muß

$$\alpha_{i-1} < \beta < \alpha_{i+1}$$

wobei außerdem $\alpha_i < \beta$ sein wird, damit mehr Objekte in der ersten Region zu liegen kommen. Beim Löschung eines Objektes aus der Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ ergeben sich somit die folgenden 3 Fälle:

Fall 1: $\text{reg}(\alpha_{i-1}, \alpha_i)$ hat nach dem Löschchen noch mindestens $1/2 M - \text{eps}$ Objekte. Dann bleiben die Region und die zugehörige Seite erhalten.

Fall 2: $\text{reg}(\alpha_{i-1}, \alpha_i)$ kann mit einer der beiden Nachbarregionen $\text{reg}(\alpha_{i-2}, \alpha_{i-1})$ oder $\text{reg}(\alpha_i, \alpha_{i+1})$ verschmolzen werden zu der neuen Region $\text{reg}(\alpha_{i-2}, \alpha_i)$ bzw. $\text{reg}(\alpha_{i-1}, \alpha_{i+1})$.

Fall 3: Die Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ wird wie im Fall 2 zunächst mit einer Nachbarregion verschmolzen, muß aber anschließend wieder gespalten werden und es entstehen die beiden Regionen $\text{reg}(\alpha_{i-2}, \beta)$ und $\text{reg}(\beta, \alpha_i)$ bzw. $\text{reg}(\alpha_{i-1}, \beta)$ und $\text{reg}(\beta, \alpha_{i+1})$.

Durch solche Lösungsvorgänge können benachbarte Regionen irgendwann wieder endgültig verschmolzen werden, so daß der FB-Baum wieder schrumpft und genau das umgekehrte Verhalten zeigt wie beim Spalten von Regionen und Seiten und dem dadurch hervorgerufenen Wachstum. Sind schließlich alle Objekte aus dem Universum gelöscht, so ist der FB-Baum wieder leer geworden.

5 Zur weiteren Erläuterung des erfindungsgemäßen Verfahrens wird nachfolgend die Beantwortung von Punktanfragen erläutert.

Bei einer Punktanfrage werden die kartesischen Koordinaten (y_1, \dots, y_n) des gesuchten Punktes P angegeben, über den dann Zusatzinformationen wie beispielsweise Höhe oder Temperatur oder Börsenwert oder dergleichen in Erfahrung gebracht werden sollen. Diese Zusatzinformationen sind mit dem Punktobjekt selbst abgespeichert.

10 Zunächst wird aus den gegebenen kartesischen Koordinaten (y_1, \dots, y_n) die Adresse pp des Punktes P berechnet. Der Punkt liegt in der eindeutig bestimmten Region $\text{reg}(\alpha_{i-1}, \alpha_i)$ mit der Eigenschaft

$$\alpha_{i-1} < pp < = \alpha_i.$$

15 Diese Region und die zu dieser Region gehörende Seite $\text{page}(\alpha_{i-1}, \alpha_i)$ wird mittels einer Suche im FB-Baum und über den dort abgelegten Zeiger pi gefunden und geholt. Auf der Seite $\text{page}(\alpha_{i-1}, \alpha_i)$ befindet sich dann die vollständige gewünschte Information über den Punkt P (und natürlich über weitere Punkte und Objekte, die zu dieser Region gehören).

20 Da ein erfindungsgemäßer FB-Baum genau wie ein B-Baum bzgl. der Höhe balanciert ist, kann die Baumsuche und somit das Auffinden des Punktes P in einer Zeit $O(\log_2 N)$ durchgeführt werden.

Eine fundamentale Anfrageart bei allen Datenbanksystemen sind sogenannte Bereichsanfragen, bei welchen bezüglich jeder Dimension ein Intervall vorgegeben wird. Keine Intervallangabe bezüglich einer Dimension wird dabei als das Intervall (- unendlich, + unendlich) betrachtet. Durch das Produkt dieser Intervalle wird ein n-dimensionaler Quader bestimmt, der den Anfragebereich darstellt. Im folgenden wird dieser Anfragebereich

25 Query-Box q genannt.

Fig. 8 zeigt beispielhaft eine Query-Box q für den 2-dimensionalen Fall, in welchem der n-dimensionalen Quader ein Rechteck ist. Der in der Fig. 8 dargestellte Anfragebereich der Query-Box q ist gegeben durch die Werte (q_{l1}, q_{l2}) für den niedrigsten Wert (l für low) und (q_{h1}, q_{h2}) für den höchsten Punkt (h für high).

30 Die Antwort auf eine Bereichsanfrage ist die Menge derjenigen Punkte oder Objekte, die in der Query-Box q liegen oder diese schneiden.

Im allgemeinen n-dimensionalen Fall ist die Query-Box gegeben durch die 2^n Werte q_{li} und q_{hi} mit $i = 1, 2, \dots, n$, wobei natürlich gemäß vorstehend erläutertem Beispiel der Fig. 8 $q_{li} < q_{hi}$ gilt (für den Fall $q_{li} = q_{hi}$ für alle i ergibt sich der Spezialfall der Punktanfrage, der schon behandelt wurde).

35 Der kleinste Punkt der Query-Box q hat somit die kartesischen Koordinaten $(q_{l1}, q_{l2}, \dots, q_{ln})$ und liegt in einer genau definierten Region $\text{reg}(\alpha_{j-1}, \alpha_j)$. Zum Auffinden dieser Region wird zunächst die Adresse

$$\lambda = \alpha(q_{l1}, q_{l2}, \dots, q_{ln})$$

40 des kleinsten Punktes von q berechnet. Dabei handelt es sich um hauptspeicherintensive Berechnungen, die keine Zugriffe zu peripheren Speichern erfordern. Der Rechenaufwand ist aus diesem Grunde vernachlässigbar klein. Dann wird die Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ mit der Eigenschaft

$$\alpha_{j-1} < \lambda < = \alpha_j$$

45 bestimmt. Das erfordert eine Suche im FB-Baum mit einem Aufwand $O(\log_2 N)$. Dabei können auch $O(\log_2 N)$ Plattenzugriffe erforderlich werden. Die letzte Seite dieser Baumsuche ist die Seite $\text{page}(\alpha_{j-1}, \alpha_j)$, die die Identifikatoren aller Datenobjekte bzw. die vollständigen Datenobjekte selbst enthält, die in der Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ liegen oder diese Region schneiden.

50 Für diese Datenobjekte wird nun einzeln bestimmt, ob sie die Query-Box q schneiden oder nicht. Es ist zu beachten, daß die Datenobjekte q nur schneiden können, wenn ihre zugehörige Region die Query-Box q schneidet (hierbei handelt es sich um eine notwendige, aber nicht hinreichende Bedingung). So wird zunächst ein Teil der Datenobjekte in der Query-Box q gefunden.

55 Die gefundene Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ ist wie vorstehend beschrieben aus Subwürfeln aufgebaut, deren Adressen geordnet sind. Als Beispiel soll hier die Region $\text{reg}(023, 101)$ aus Fig. 7 dienen, die aus den Würfeln 024, 04, 101 in dieser Reihenfolge besteht. Wenn eine Region den Anfragebereich schneidet, müssen natürlich nicht alle Subwürfel dieser Region den Anfragebereich ebenfalls schneiden.

60 Die Adresse des letzten Subwürfels der Region $\text{reg}(\alpha_{j-1}, \alpha_j)$, der den Anfragebereich schneidet, sei β . Außerdem habe β die Form $\beta' l$ wobei l der Index von β auf der Stufe ist, auf der sich β befindet. Beispielsweise gilt für den Würfel 024 der Region $\text{reg}(023, 101)$ der Fig. 7:

für 024 auf der Stufe 3 ist $l = 4$.

65 $\beta = 024$ läßt sich somit in der Form $\beta' l$ mit $\beta' = 02$ und $l = 4$ darstellen, wobei die Anzahl der Ziffern in der Adressdarstellung die Stufe wiedergibt.

Dasselbe gilt für die weiteren in der genannten Region $\text{reg}(023, 101)$ vorhandenen Würfel 04 und 101:

für 04 auf der Stufe 2 ist $l = 4$

für 101 auf der Stufe 3 ist $l = 1$.

Es ist zu beachten, daß $l = 0$ nicht vorkommen kann, da nach der erfundungsgemäßen Konstruktion der Adressen keine Adresse mit 0 endet.

Nachdem die Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ wie vorstehend beschrieben bei der Bereichsanfrage abgearbeitet ist, muß nun die nächste Region gefunden werden, die den Anfragebereich schneidet. Dazu wird die Lage von Würfel β im Bezug zu der Query-Box q und zu seinem übergeordneten Würfel, in dem β selbst enthalten ist, betrachtet. Dies ist beispielhaft in der Darstellung der Fig. 9 gezeigt, in welcher die Query-Box q der Fig. 8 mit darin liegendem Würfel β dargestellt ist, wobei die weiteren Würfel, die zu dem übergeordneten Würfel des Würfels β gehören, punktiert eingezeichnet sind. Diese punktiert eingezeichneten Subwürfel der gleichen Stufe, die zu einem selben übergeordneten Würfel gehören, werden hier als Brüder bezeichnet. Alle Subwürfel der Stufe s eines übergeordneten Würfels der Stufe $s-1$ sind somit Brüder. Ein Subwürfel derselben Stufe s mit einem größeren Index l ist damit ein kleinerer Bruder, ein Subwürfel derselben Stufe s mit einem größeren Index l dabei ein größerer Bruder. In dem in Fig. 7 dargestellten Beispiel ist der große Bruder des Subwürfels 023 der Subwürfel 024, wobei diese beiden Brüder nicht in derselben Region liegen.

Bei der Weiterführung der Bereichsanfrage sind somit die nächsten Datenobjekte, die bisher in der schon abgearbeiteten Region $\text{reg}(\alpha_{j-1}, \alpha_j)$ noch nicht gefunden wurden und in der Query-Box q liegen, in einem größeren Bruder von β oder im Vater von β oder in einem anderen Vorfahren von β enthalten.

Fig. 10 zeigt beispielhaft einige mögliche Situationen für einen Würfel β der Form β'^2 für den 2-dimensionalen Fall.

Falls sich kein größerer Bruder des Würfels β mit der Query-Box q schneidet (letzter Fall in Fig. 10), dann enthält auch der Vater des Würfels β keine Objekte mehr, die noch nicht gefunden sind (durch vorherige Abarbeitung der kleineren Brüder), aber in der Query-Box q liegen könnten. Es müssen deshalb die größeren Brüder des Vaters des Würfels β dahingehend untersucht werden, ob sie die Query-Box q schneiden. Falls nicht, muß nach einer analogen Überlegung zum Großvater des Würfels β übergegangen und dessen größere Brüder auf Überschneidung mit der Query-Box q geprüft werden etc. Auf diese Weise wird schließlich das ganze Datenuniversum abgedeckt und alle Objekte in der Query-Box q werden gefunden.

Man beachte: Wenn sich der Würfel β auf der Stufe s befindet, dann können wir höchstens s -mal zum Vater übergehen und jeweils die größeren Brüder (davon gibt es höchstens $2^s - 1$) auf Überschneidung mit der Query-Box q prüfen. Außerdem ist $s \leq l_d(\text{pix})$ und beim Übergang zum Vaterknoten sowie bei der Prüfung der Überschneidung der jeweils größeren Brüder mit der Query-Box q werden nur hauptspeicherinterne Berechnungen durchgeführt; es sind also keine Ein/Ausgabevorgänge oder Plattenzugriffe nötig. Deshalb ist dieses Verfahren, den nächsten Würfel zu finden, der die Query-Box q schneidet, extrem schnell und kann in der Bilanz für die Gesamtzeit bei der Beantwortung einer Bereichsanfrage vernachlässigt werden.

Sobald nach diesem Verfahren der erste Würfel bestimmt wurde, der die Query-Box q schneidet, werden die kartesischen Koordinaten des kleinsten Pixels im Durchschnitt mit der Query-Box q (in Fig. 10 die kleinen schwarzen Quadrate) berechnet, die sich wie folgt ergeben:
Ein Würfel habe bzgl. Dimension i die Ausdehnung von x_{l_i} (kleinste Koordinate) bis x_{h_i} (größte Koordinate) in Anlehnung an die Bezeichnungen für die Werte q_l bzw. q_h .

Die Bedingung dafür, daß dieser Würfel die Query-Box q nicht schneidet, ist:

es existiert $i: x_{h_i} < q_l$ or $x_{l_i} > q_h$

Die Bedingung dafür, daß dieser Würfel die Query-Box q schneidet, ist die Negation obiger Formel:

not es existiert $i: x_{h_i} < q_l$ or $x_{l_i} > q_h$

bzw. nach den Gesetzen der mathematischen Logik:

for all $i: x_{h_i} \geq q_l$ and $x_{l_i} \leq q_h$

Dann ergeben sich die Koordinaten des kleinsten Schnittpunktes sp mit der Query-Box q wie folgt bzgl. der i -ten Dimension:

if $x_{l_i} \geq q_l$ then $sp_i = x_{l_i}$ else $sp_i = q_l$

Der Schnittpunkt sp hat dann die kartesischen Koordinaten

$sp = (sp_1, sp_2, \dots, sp_n)$

und seine Adresse ist:

$\text{sigma} = \text{alpha}(sp_1, sp_2, \dots, sp_n)$

Man beachte, daß bis zu diesem Schritt unser Verfahren zur Bestimmung von sp keine Ein/Ausgabevorgänge oder Plattenzugriffe benötigte.

Um die zu dem Schnittpunkt sp gehörende eindeutige Region zu finden ist nun eine Punktanfrage erforderlich mit der Adresse sigma , genau wie sie oben schon beschrieben wurde. Sie erfordert einen Zeitaufwand von der

Ordnung $O(\log kN)$, wie oben ebenfalls schon analysiert wurde.

Daraus folgt nun aber, daß bei der Beantwortung einer Bereichsanfrage nur Bearbeitungskosten für diejenigen Regionen anfallen, die die Query-Box q tatsächlich schneiden. Für jede solche Region sind die Kosten von der Ordnung $O(\log kN)$, insgesamt also $r^*O(\log kN)$ Kosten, wenn r Regionen die Query-Box q schneiden.

Nachfolgend wird das Verfahren zur Beantwortung von Bereichsanfragen für eine n -dimensionale Query-Box q mit Koordinaten q_{l_i} und q_{h_i} für $i = 1, 2, \dots, n$ angegeben:

Initialize:

10 $\sigma := \alpha(q_{l_1}, q_{l_2}, \dots, q_{l_n});$

RegionLoop: begin co für jede Region, die q schneidet oc

15 finde durch Baumsuche im FB-Baum die Region $reg(\alpha_{j-1}, \alpha_j)$,

 in der σ liegt, d.h. mit $\alpha_{j-1} < \sigma \leq \alpha_j$;

 hole Seite $page(\alpha_{j-1}, \alpha_j);$

20 ObjectLoop: begin für alle Objekte Q auf Seite $page(\alpha_{j-1}, \alpha_j)$

prüfe:

 if Q intersects q then Ausgabe von Q als Teil der Antwort

end ObjectLoop;

25 finde letzten Subcube mit Adresse β von $reg(\alpha_{j-1}, \alpha_j)$,

 so daß $Subcube(\beta)$ intersects q :

30 if $(qh_1, qh_2, \dots, qh_n)$ contained in $Subcube(\beta)$ then co fertig oc goto

Exit else

FatherLoop: begin co β habe die Form $\beta = \beta'.i$ oc

35 $i := tail(\beta);$

BrotherLoop: for $k := i+1$ to 2^n

40 do if $Subcube(\beta'.k)$ intersects q then

begin $sp :=$ smallest intersection with q ;

$\sigma := \alpha(sp);$

goto RegionLoop

45 co Schleife wird hier sicher verlassen,
 weil q noch nicht abgearbeitet ist oc

end

od co für alle größeren Brüder von β ist Durchschnitt mit

50 q leer oc;

$\beta := father(\beta);$

goto FatherLoop

55 end FatherLoop;

end RegionLoop;

60 Exit: co Ende des Programms ec

Nachfolgend wird das erfundungsgemäße Verfahren beispielhaft anhand der Verwaltung von allgemeinen ausgedehnten Objekten beschrieben.

65 Ein allgemeines ausgedehntes Objekt ist ein wichtiger Fall eines Datenobjekts. Dabei handelt es sich beispielsweise um einen See in einer geographischen Karte, wie er in Fig. 11 dargestellt ist.

Das ausgedehnte Objekt O wird zuerst durch einen der Dimension entsprechenden achsenparallelen Quader umgeben, der in seinen Abmessungen so klein gewählt wird, daß er das ausgedehnte Objekt gerade umgibt. In

dem Beispiel der Fig. 11 ist dies an dem dort abgebildeten See und einem den See gerade umgebenden 2-dimensionalen Quader bb (= Rechteck) dargestellt. In der Fachliteratur wird der das ausgedehnte Objekt umgebende Quader als Bounding Box bezeichnet.

für ein ausgedehntes Objekt wird im FB-Baum lediglich der Identifikator $Id(O)$ abgespeichert, wobei diese Abspeicherung des Identifikators im allgemeinen sogar mehrfach erfolgt, nämlich für jede Region, die das ausgedehnte Objekt schneidet. Das ausgedehnte Objekt selbst ist aus dem FB-Baum in einen anderen Speicherbereich oder in eine Datenbank ausgelagert.

Die Bounding Box für ein ausgedehntes Objekt O wird hier mit $bb(O)$ bezeichnet. Der Identifikator $Id(O)$ für das ausgedehnte Objekt O wird nun in dem FB-Baum mit jeder Region gespeichert, die das ausgedehnte Objekt O schneidet. Dabei ist zu beachten, daß das ausgedehnte Objekt O nur diejenigen Regionen schneiden kann, die auch von der Bounding Box bb geschnitten werden. Dies ist eine notwendige, jedoch keine hinreichende Bedingung, die eingesetzt wird, um die Algorithmen zur Umsetzung des erfindungsgemäßen Verfahrens wesentlich zu beschleunigen.

Beim Einfügen eines allgemeinen ausgedehnten Objekts O wird zuerst die zugehörige Bounding Box $bb(O)$ berechnet. Dann wird das folgende Verfahren ausgeführt:

5

10

15

20

25

30

35

40

45

50

55

60

65

for all Regionen R, die bb(O) schneiden do
 if R intersects O then fuge Id(O) in R ein
 co dies kann natürlich zu einer, i.a. sogar zu mehreren Spaltungen von R führen oc

5

Hinweis: Um die Regionen R zu finden, die bb(O) schneiden, behandelt man bb(O) genau wie eine Query Box q. Das führt zu folgendem detaillierten Verfahren:

10

15

Initialize:

berechne bb(O);

q := bb(O);

sigma := alpha (ql₁ , ql₂ , ..., ql_n) ;

20

25

RegionLoop: begin co für jede Region, die q schneidet oc

finde durch Baumsuche im FB-Baum die Region reg (α_{j-1}, α_j),
 in der sigma liegt, d.h. mit α_{j-1} << sigma <<= α_j ;hole Seite page (α_{j-1}, α_j);

if O intersects R then fuge Id(O) in R ein, d.h.:

if Anzahl der Objekte, die R schneiden, ist <= M
 then speichere Id(O) auf page (α_{j-1}, α_j)

30

else spalte R und page (α_{j-1}, α_j)wie vorstehend zum Spalten von Regionen und Seiten
 beschrieben;

35

finde letzten Subwürfel mit Adresse β von reg (α_{j-1}, α_j),
 so daß Subcube (β) intersects q;

40

if (q_{h₁}, q_{h₂} , ..., q_{h_n}) contained in Subcube (β) then co fertig ec goto Exit
 else

Father Loop: begin co β habe die Form β = β'.i oc

i := tail (β) ;

45

BrotherLoop: for k := i+1 to 2ⁿ

do if Subcube (β'.k) intersects q then

begin sp := smallest intersection with q;

50

sigma := alpha (sp);

goto RegionLoop

co Schleife wird hier sicher verlassen,
 weil q noch nicht abgearbeitet ist ec

55

end

od co für alle größeren Brüder von β ist Durchschnitt mit
 q leer oc ;

60

β := father (β);

goto FatherLoop

end FatherLoop;

65

end RegionLoop ;

Exit: co Ende des Programms oc

Zum Löschen eines ausgedehnten Objektes O wird wieder eine Bounding Box $bb(O)$ verwendet, um alle Regionen zu finden die O schneiden können. Der Identifikator $Id(O)$ ist in den Regionen enthalten und auf den zugehörigen Seiten gespeichert, die das ausgedehnte Objekt O tatsächlich schneiden, und wird daraus gelöscht. Dabei kann es wieder zum Verschmelzen von Regionen und zugehörigen Seiten kommen, wie es vorstehend schon beschrieben wurde.

Das erfundungsgemäße Datenbanksystem und die erfundungsgemäßen Verfahren zur Verwaltung multidimensionaler Daten gestatten somit einen schnellen und sicheren Zugriff auf Daten eines multidimensionalen Datenbestands, wobei die erfundungsgemäße Datenstruktur insbesondere ein dynamisches Ergänzen oder Ändern des multidimensionalen Datenbestands erlaubt. Erfundungsgemäß ist lediglich die Modifikation eines einzelnen Baumes für das Einfügen bzw. Löschen von Objekten notwendig.

Bei der Beantwortung von Anfragen zeigt das erfundungsgemäße FB-Baum-Verfahren folgende Leistungsmerkmale: Wir nehmen dazu an, daß $p_i\%$ der Werte des Datenbestandes bezüglich der i-ten Dimension in dem Anfrageintervall $(q_i; q_{i+1})$ der Query-Box q liegen, dann liegen in der Query-Box q

$$(p_1\% * p_2\% * \dots * p_n\%)^* N$$

Objekte. Da nicht alle Regionen, die q schneiden, vollständig innerhalb von q liegen, sondern darüber hinausragen können, müssen i.a. mehr Objekte vom peripheren Speicher geholt werden, als in der Query-Box liegen. Im Mittel sind das aber weniger als zweimal soviel Objekte, wie in q liegen, d. h. $2^*(p_1\% * p_2\% * \dots * p_n\%)^* N$ Objekte.

Es liegt somit ein multiplikatives anstatt ein additives Verhalten (wie bei den aus dem Stand der Technik bekannten Verfahren) von Bruchteilen von N vor, was zu deutlichen Verbesserungen führt. Dies wird anhand eines einfachen Rechenbeispiels illustriert:

Seien $p_1 = 2\%$, $p_2 = 5\%$, $p_3 = 4\%$, $p_4 = 10\%$
dann ist die Summe der $p_i = 21\% = 21 \cdot 10^{-2}$
und das Produkt der $p_i = 400 \cdot 10^{-8} = 4 \cdot 10^{-6}$.

Wenn das gesamte betrachtete Datenuniversum 10.000.000 Objekte enthält – für typische Datenbankanwendungen ein realistisches, eher kleines Datenuniversum – dann müssen mit dem derzeitigen Stand der Technik 2100.000 Objekte vom peripheren Speicher geholt werden, mit dem neuen Verfahren der FB-Bäume aber nur $2 \cdot 10^7 \cdot 40^{-6} = 80$ Objekte, eine Verbesserung etwa um den Faktor 2500 gegenüber dem bekannten Stand der Technik.

Selbstverständlich ist die vorliegende Erfahrung nicht auf die beschriebenen Ausführungsformen beschränkt, sondern es sind andere Ausgestaltungen möglich, die im Bereich des fachmännischen Könnens liegen. So ist beispielsweise die Art und Weise der Numerierung der Subwürfel und der Aufbau der Adressen von Subwürfeln und Regionen auch auf andere Weise möglich, ohne daß der Bereich der Erfahrung verlassen wird.

Patentansprüche

1. Datenbanksystem mit einer Recheneinrichtung, einem Hauptspeicher und einer insbesondere peripheren Speichereinrichtung, in dem zum Indizieren und Speichern eines in einem mehrdimensionalen Würfel liegenden Datenbestands auf Speicherseiten gegebener Speicherkapazität des insbesondere peripheren Speichers mitteils ein wiederholtes iteratives Unterteilen des mehrdimensionalen Würfels in allen Dimensionen in Subwürfel erfolgt bis aufeinanderfolgende Subwürfel zu Regionen zusammenfaßbar sind die jeweils eine Menge von Datenobjekten beinhalten, die auf einer der Speicherseiten gegebener Speicherkapazität abspeicherbar sind.
2. Datenbanksystem nach Anspruch 1, in dem das Abspeichern der Datenobjekte einer Region auf einer Speicherseite gegebener Speicherkapazität unter Zuordnung eines Zeigers zu der Speicherseite und einer die Regionengrenzen bezeichnenden Adresse erfolgt.
3. Datenbanksystem nach Anspruch 2, in dem die Speicherung des Zeigers und der Adresse in einem B-Baum, B*-Baum oder Präfix-B-Baum erfolgt.
4. Datenbanksystem nach Anspruch 3, in dem die Speicherung der Datenobjekte in den Blattseiten des B-Baums, B*-Baums oder Präfix-B-Baum erfolgt.
5. Datenbanksystem nach einem der Ansprüche 2 bis 4, wobei die die Regionengrenzen bezeichnende Adresse aus Angaben über den letzten der die Region bildenden Subwürfeln besteht.
6. Datenbanksystem nach Anspruch 5, in dem die Adresse Angaben über die Anzahl der auf jeder Unterteilungsstufe in der Region enthaltenen Subwürfel umfaßt.
7. Datenbanksystem nach einem der Ansprüche 5 oder 6, in dem die Adresse aus einer Abfolge von dem jeweils letzten auf der jeweiligen Unterteilungsstufe noch von der Region vollständig eingeschlossenen Subwürfe zugeordneten Nummern besteht.
8. Verfahren zum Verwalten eines in einem n-dimensionalen Würfel mit $n > 1$ liegenden Datenbestands, wobei zum Indizieren und Speichern des Datenbestands auf Speicherseiten gegebener Speicherkapazität einer insbesondere peripheren Speichereinrichtung die folgenden Schritte ausgeführt werden:
 1. Bilden einer Ausgangsregion bestehend aus dem Gesamtwürfel,
 2. Zählen der in der Ausgangsregion vorhandenen Datenobjekte,
 3. Abspeichern der Datenobjekte der Ausgangsregion auf einer Speicherseite, falls die Anzahl der Datenobjekte der Ausgangsregion nicht größer ist als die der vorgegebenen Speicherkapazität einer Speicherseite entsprechende Zahl von Datenobjekten,

4. andernfalls Auswählen einer Spaltadresse für die Ausgangsregion derart, daß durch Spalten der Ausgangsregion entlang der Spaltadresse eine erste und eine zweite Teilregion erzeugt werden, in welchen jeweils etwa die Hälfte der Datenobjekte der Ausgangsregion enthalten sind.

5. Fortführen des Verfahrens ab Schritt 2, wobei die erste Teilregion die Rolle der Ausgangsregion übernimmt.

6. Durchführen des Verfahrens ab Schritt 2, wobei die zweite Teilregion die Rolle der Ausgangsregion übernimmt.

7. Verfahren nach Anspruch 6, wobei das Abspeichern gemäß Schritt 3 unter partieller Abspeicherung einer der betreffenden Region zugeordneten Adresse sowie eines der Adresse zugeordneten und auf die Speicherseite, auf der die gemäß Schritt 3 abgespeicherten Datenobjekte enthalten sind, verweisenden Zeigers erfolgt.

8. Verfahren nach Anspruch 7, bei dem die Speicherung der Adresse und des Zeigers in einem B-Baum, B*-Baum oder Präfix-B-Baum erfolgt, wobei durch aufeinanderfolgende Adressen jeweils Regionen definiert werden, deren Datenobjekte jeweils auf einer Speicherseite gegebener Speicherkapazität abgespeichert sind.

9. Verfahren zum Einfügen von Datenobjekten in einen insbesondere nach einem Verfahren gemäß einem der Ansprüche 8 bis 10 in Regionen aufgeteilten und auf Speicherseiten gegebener Speicherkapazität einer insbesondere peripheren Speichereinrichtung abgespeicherten n-dimensionalen Datenbestand mit $n > 1$ mit den folgenden Schritten:

10. Bestimmen einer dem einzufügenden Datenobjekt zugehörigen Region des n-dimensionalen Datenbestands aus den Koordinaten des Datenobjekts im n-dimensionalen Raum.

11. Auffinden der Speicherseite, auf der diese Region abgespeichert ist,

12. Bestimmen der Anzahl der auf dieser Speicherseite gespeicherten Datenobjekte,

13. Speichern des Datenobjekts auf dieser Speicherseite falls die Anzahl der gespeicherten Datenobjekte kleiner ist als die der gegebenen Speicherkapazität der Speicherseite entsprechende Anzahl,

14. andernfalls Auswählen einer Spaltadresse für die auf dieser Speicherseite gespeicherte Region derart daß durch Spalten der Region entlang der Spaltadresse eine erste und eine zweite Teilregion erzeugt werden in welchen jeweils weniger als etwa die Hälfte der der gegebenen Speicherkapazität entsprechenden Anzahl an Datenobjekten enthalten ist,

15. Einfügen des Datenobjekts in diejenige Teilregion, in der die Koordinaten des Datenobjekts liegen,

16. Speichern der ersten und der zweiten Teilregion auf jeweils einer Speicherseite.

17. Verfahren nach Anspruch 11, bei dem das Auffinden der Speicherseite gemäß Schritt 2 mittels in einem B-Baum, B*-Baum oder Präfix-B-Baum gespeicherter, den Speicherseiten zugeordneter Adressen und Zeiger erfolgt.

18. Verfahren nach Anspruch 12, bei dem das Speichern der ersten und der zweiten Teilregion gemäß Schritt 7 unter Ersetzung des Zeigers und der Adresse der Region aus Schritt 2 durch jeweils der ersten und der zweiten Teilregion zugeordnete Adressen und Zeiger erfolgt.

19. Verfahren nach Anspruch 13, bei dem die Speicherung der Adresse und des Zeigers in einem B-Baum, B*-Baum oder Präfix-B-Baum erfolgt, wobei durch aufeinanderfolgende Adressen jeweils Regionen definiert werden, deren Datenobjekte jeweils auf einer Speicherseite gegebener Speicherkapazität abgespeichert sind.

20. Verfahren zum Löschen von Datenobjekten in einem insbesondere nach einem Verfahren gemäß einem der Ansprüche 8 bis 10 in Regionen aufgeteilten und auf Speicherseiten gegebener Speicherkapazität einer insbesondere peripheren Speichereinrichtung abgespeicherten n-dimensionalen Datenbestand mit den folgenden Schritten:

21. Bestimmen einer dem zu löschenen Datenobjekt zugehörigen Region des n-dimensionalen Datenbestands aus den Koordinaten des Datenobjekts im n-dimensionalen Raum,

22. Auffinden der Speicherseite, auf der diese Region abgespeichert ist,

23. Löschen des Datenobjekts auf dieser Speicherseite,

24. Bestimmen der Anzahl der auf dieser Speicherseite gespeicherten Datenobjekte,

25. Verschmelzen der Region mit einer ihrer beiden Nachbarregionen falls die Anzahl der gespeicherten Datenobjekte kleiner ist als etwa die Hälfte der der gegebenen Speicherkapazität der Speicherseite entsprechenden Anzahl,

26. Bestimmen der Anzahl der in der durch das Verschmelzen neu entstandenen Region vorhandenen Datenobjekte,

27. Speichern der Region auf einer Speicherseite falls die Anzahl der in der Region vorhandenen Datenobjekte kleiner ist als die der gegebenen Speicherkapazität der Speicherseite entsprechende Anzahl,

28. andernfalls Auswählen einer Spaltadresse für die Region derart, daß durch Spalten entlang der Spaltadresse eine erste Teilregion und eine zweite Teilregion erzeugt werden, die jeweils etwa die Hälfte der gemäß Schritt 6 bestimmten Datenobjekte enthalten und Speichern der Teilregionen auf jeweils einer Speicherseite.

29. Verfahren nach Anspruch 15, wobei das Auffinden der Speicherseite gemäß Schritt 2 mittels in einem B-Baum, B*-Baum oder Präfix-B-Baum gespeicherter, den Speicherseiten zugeordneter Adressen und Zeiger erfolgt.

30. Verfahren zum Durchführen einer Datenanfrage auf der Grundlage eines gegebenen a-dimensionalen Anfragebereichs in einem insbesondere nach einem Verfahren gemäß einem der Ansprüche 8 bis 10 in aus Subwürfeln zusammengesetzte Regionen aufgeteilten und auf Speicherseiten gegebener Speicherkapazität

einer insbesondere peripheren Speichereinrichtung abgespeicherten n-dimensionalen Datenbestand mit den folgenden Schritten:

1. Bestimmen der Koordinaten des niedrigsten und des höchsten Schnittpunktes des Anfragebereichs mit dem n-dimensionalen Datenbestand.
2. Bestimmen derjenigen Region, in welcher der niedrigste Schnittpunkt liegt und Auffinden der Speicherseite, auf der diese Region abgespeichert ist, 5
3. Ermitteln sämtlicher auf dieser Speicherseite abgespeicherter Datenobjekte, die eine Schnittmenge mit dem Anfragebereich bilden und Ausgabe der ermittelten Datenobjekte.
4. Bestimmen des in der Abfolge letzten Subwürfels der in Schritt 2 bestimmten Region, der den Anfragebereich schneidet, 10
5. Beenden der Datenanfrage falls der höchste Schnittpunkt in diesem Subwürfel liegt,
6. andernfalls Ermitteln des nächsten Subwürfels derselben Ebene und desselben übergeordneten Würfels, der den Anfragebereich schneidet,
7. Bestimmen der Koordinaten des niedrigsten Schnittpunktes des Anfragebereichs mit dem ermittelten Subwürfel und Weiterführen des Verfahrens bei Schritt 2, falls in Schritt 6 ein Subwürfel ermittelt 15 wurde,
8. andernfalls Ermitteln des nächsten Subwürfels der Ebene des übergeordneten Würfels aus Schritt 6, der den Anfragebereich schneidet und Durchführen von Schritt 6 mit den Subwürfeln des ermittelten Würfels,
9. Fortsetzen des Verfahrens mit Schritt 6, falls in Schritt 8 kein Subwürfel der Ebene des übergeordneten Würfels aus Schritt 6 ermittelt wurde, wobei der übergeordnete Würfel die Rolle des Subwürfels übernimmt, 20

Hierzu 5 Seite(n) Zeichnungen

25

30

35

40

45

50

55

60

65

- Leerseite -

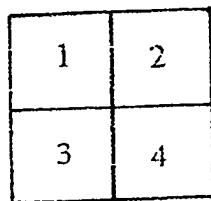


FIG. 1

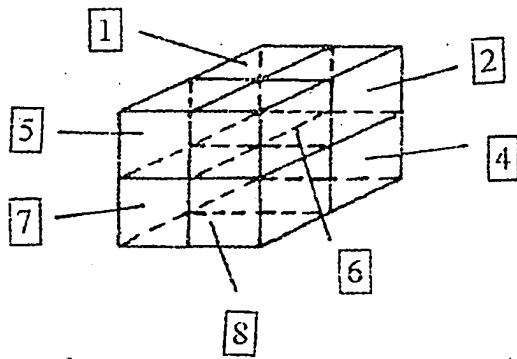


FIG. 2

FIG. 3.1

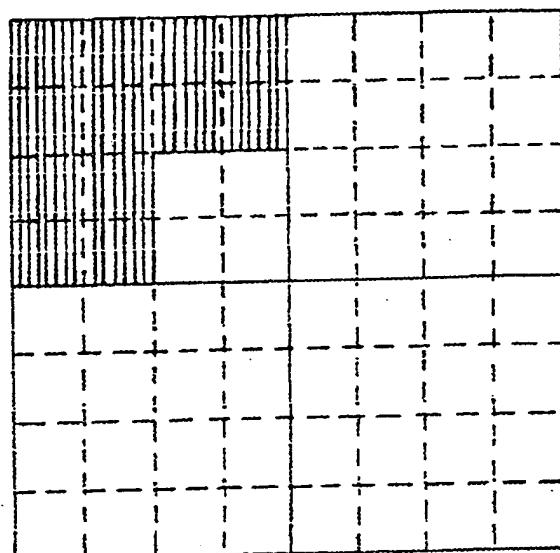


FIG. 3.2

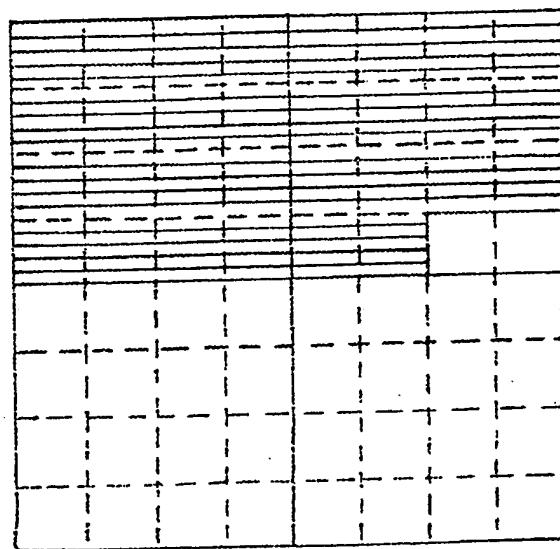


FIG. 3.3

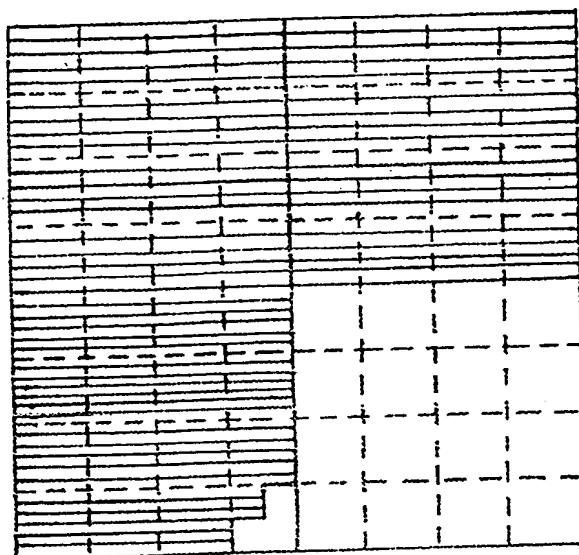
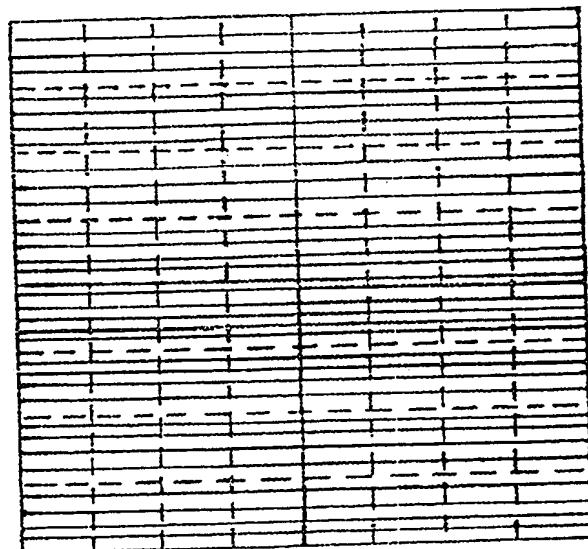


FIG. 3.4



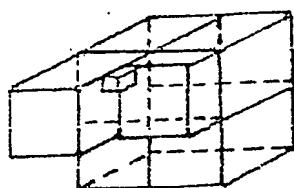


FIG. 4

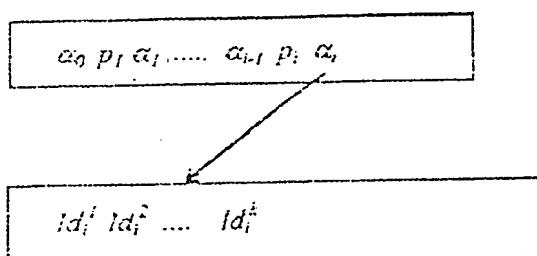


FIG. 5

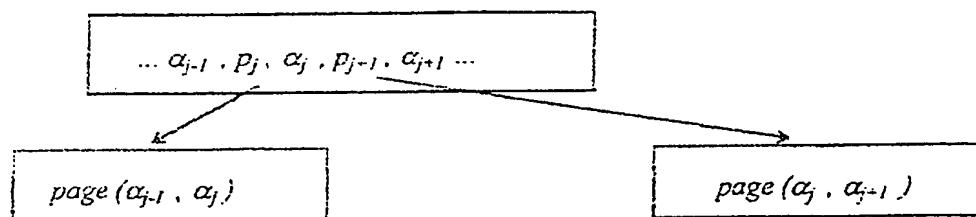


FIG. 6.1

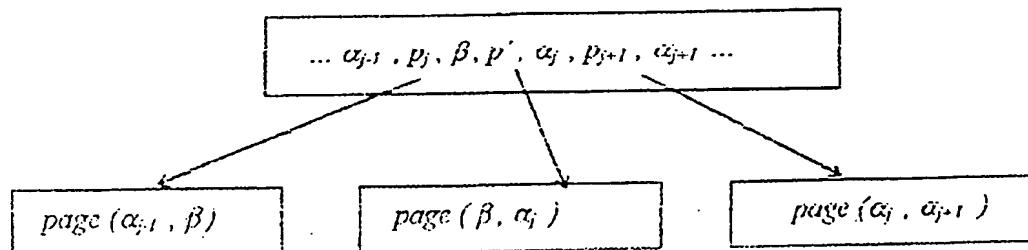


FIG. 6.2

FIG. 7

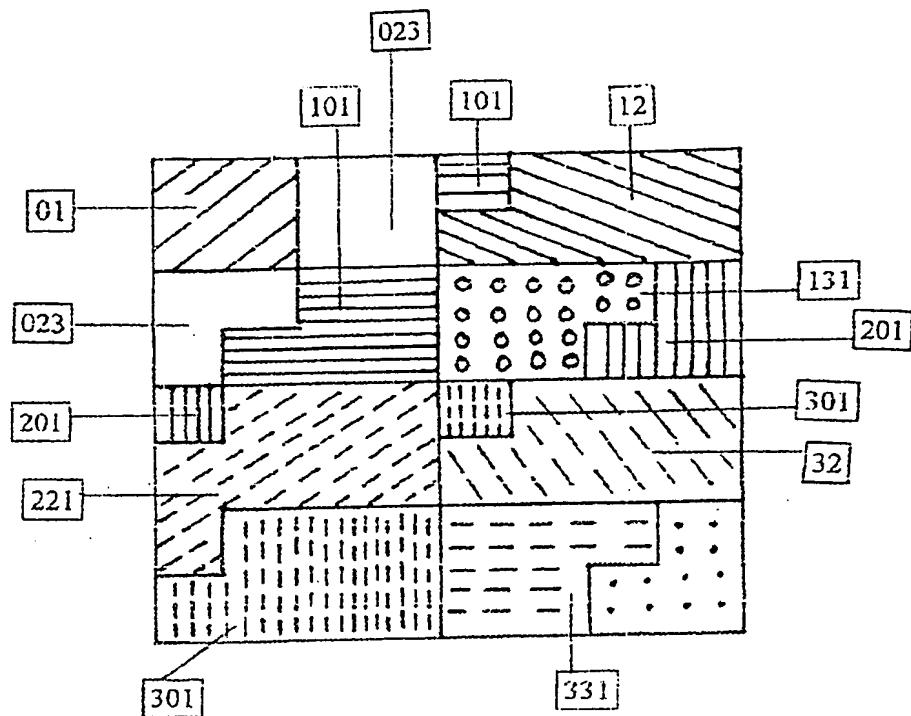


FIG. 8

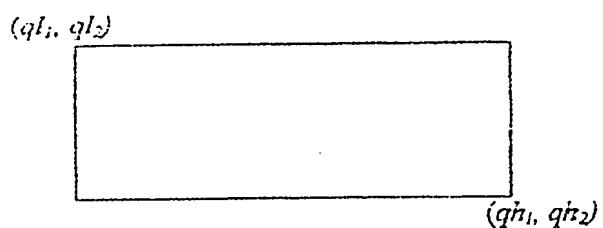
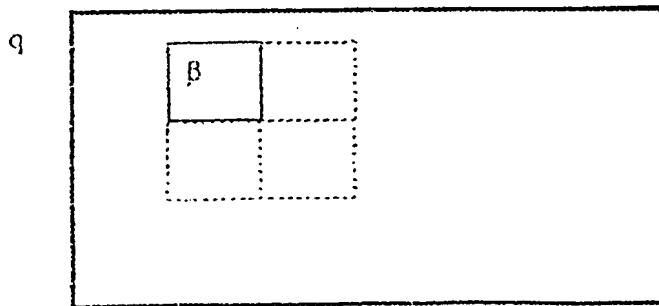


FIG. 9



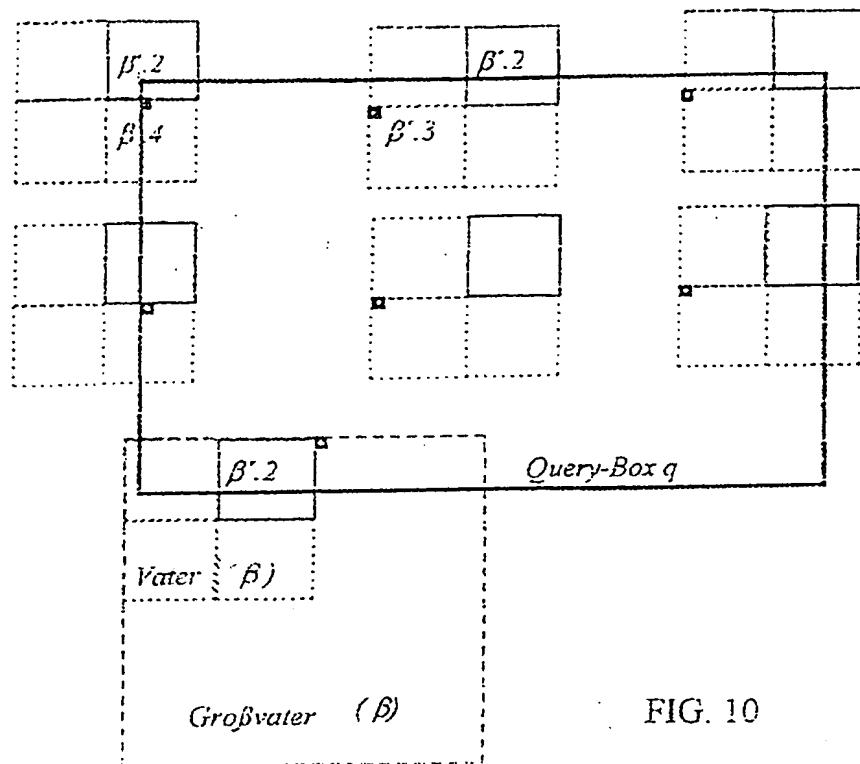


FIG. 10

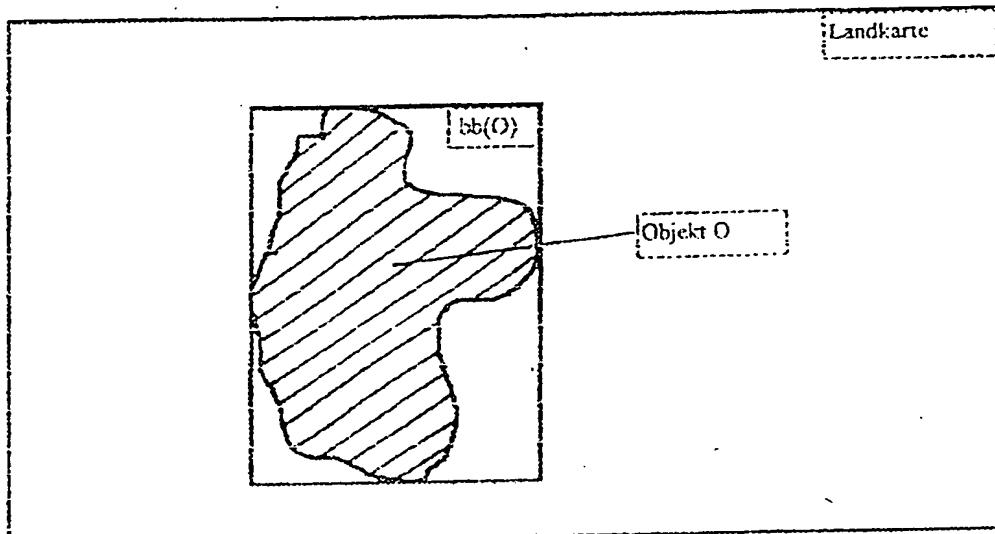


FIG. 11

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.